凸优化和单调变分不等式的收缩算法

第九讲: 求解线性约束凸优化
基于对偶上升的自适应方法

Self-adaptive dual ascent method for
linearly constrained convex optimization

南京大学数学系　　何炳生
hebma@nju.edu.cn

1 Introduction

Let $\mathcal{X}$ be a convex closed set in $\Re^n$. The problem concerned in this note is the constrained convex minimization problem

$$\text{(P)} \qquad \min \{f(x) \,|\, Ax = b,\, x \in \mathcal{X}\}, \tag{1.1}$$

where $f(x) : \Re^n \to \Re$ is a differentiable convex function, $A \in \Re^{m \times n}$ and $b \in \Re^m$. Let λ be the Lagrange multiplier to the linear constraints $Ax = b$, the Lagrange function of the problem (1.1) is

$$L(x, \lambda) = f(x) - \lambda^T (Ax - b), \tag{1.2}$$

which is defined on $\mathcal{X} \times \Re^m$. The dual problem of (1.1) is

$$\max \phi(\lambda),$$

where

$$\phi(\lambda) = \inf_{x \in \mathcal{X}} L(x, \lambda).$$

Assuming that strong duality holds, the optimal values of the primal and dual problems are the same. We can recover a primal optimal point x^* from a dual optimal point λ^* by

solving

$$x^* = \arg\min\{L(x, \lambda^*) | x \in \mathcal{X}\},$$

provided there is only one minimizer of $L(x, \lambda^*)$ (this is the case if, for example, f is strictly convex).

The dual problem can also be interpreted as

$$\text{(D)} \qquad \begin{aligned} \max_{x,\lambda} \quad & L(x, \lambda) = f(x) - \lambda^T (Ax - b) \\ \text{s. t} \quad & x \in \mathcal{X},\ (x' - x)^T \nabla_x L(x, \lambda) \geq 0,\ \forall\, x' \in \mathcal{X}. \end{aligned} \tag{1.3}$$

We denote the solution set of (1.3) by $\Omega^* = \mathcal{X}^* \times \Lambda^*$.

Dual feasible pair A pair of (x, λ) is dual feasible of (1.3) if and only if

$$x \in \mathcal{X},\ (x' - x)^T (\nabla f(x) - A^T \lambda) \geq 0,\ \forall\, x' \in \mathcal{X}. \tag{1.4}$$

For any given $\lambda^k \in \Re^m$, we let x^k be defined as in (1.5a). Therefore, (x^k, λ^k) is a feasible solution of (1.3). Note that (x^*, λ^*) is also dual feasible.

The dual ascent method is the algorithm

$$x^k := \arg\min\{L(x, \lambda^k) | x \in \mathcal{X}\}, \tag{1.5a}$$

and then

$$\lambda^{k+1} = \lambda^k - \alpha_k(Ax^k - b), \tag{1.5b}$$

where $\alpha_k > 0$ is the step size and will be discussed later. The first step (1.5a) is an x-minimization step, and the second step (1.5b) is a dual variable update. In some practical applications, the dual variable λ can be viewed as a vector of prices, and the λ-update step is called a price update or price adjustment step. It is called dual ascent since, with appropriate choice of α, the dual function increases in each step, i.e.,

$$\phi(\lambda^{k+1}) > \phi(\lambda^k).$$

For any dual feasible pair (x, λ), because

$$L(x^*, \lambda^*) = f(x^*) \geq f(x) + (x^* - x)^T \nabla f(x)$$

and

$$(x^* - x)^T (\nabla f(x) - A^T \lambda) \geq 0,$$

it follows that

$$L(x^*, \lambda^*) \geq L(x, \lambda).$$

Thus, $L(x^*, \lambda^*)$ is the maximal value of the dual problem (1.3). By setting

$$v_k = L(x^*, \lambda^*) - L(x^k, \lambda^k), \tag{1.6}$$

the sequence $\{v_k\}$ is non-negative. This paper considers the convergence rate of the non-negative sequence $\{v_k\}$ by the dual ascent method.

Assumption Throughout this paper, we assume that the function $f(x)$ is uniformly strict convex. In other words, there is a positive constant $\mu > 0$, such that

$$(x - \tilde{x})^T \big(\nabla f(x) - \nabla f(\tilde{x})\big) \geq \mu \|x - \tilde{x}\|^2, \quad \forall x, \tilde{x}. \tag{1.7}$$

Lemma 1.1 *Let (x, λ) and $(\tilde{x}, \tilde{\lambda})$ be any given dual feasible pairs. Under the assumption* (1.7)*, we have*

$$\|x - \tilde{x}\| \leq \frac{1}{\mu} \|A^T(\lambda - \tilde{\lambda})\|. \tag{1.8}$$

Proof. Since (x, λ) and $(\tilde{x}, \tilde{\lambda})$ be any given dual feasible pairs, thus we have

$$x \in \mathcal{X}, \quad (\tilde{x} - x)^T \big(\nabla f(x) - A^T \lambda\big) \geq 0,$$

and

$$\tilde{x} \in \mathcal{X}, \quad (x - \tilde{x})^T \big(\nabla f(\tilde{x}) - A^T \tilde{\lambda}\big) \geq 0.$$

Adding the above two inequalities, we obtain

$$(x - \tilde{x})^T A^T (\lambda - \tilde{\lambda}) \geq (x - \tilde{x})^T \big(\nabla f(x) - \nabla f(\tilde{x})\big) \geq \mu \|x - \tilde{x}\|^2,$$

and it follows the assertion (1.8) directly. □

In other words, under the assumption that f is uniformly strict convex and differentiable, the solution of (1.5a) is a Lipschitz continuous function of λ.

Lemma 1.2 *For given λ^k, let x^k be given by* (1.5a)*. Then for any feasible solution (x, λ) of the dual problem* (1.3)*, we have*

$$L(x^k, \lambda^k) - L(x, \lambda) \geq (\lambda - \lambda^k)^T (Ax^k - b). \tag{1.9}$$

Proof. First, using the convexity of f we obtain

$$\begin{aligned} & L(x^k, \lambda^k) - L(x, \lambda) \\ = \quad & f(x^k) - f(x) + \lambda^T (Ax - b) - (\lambda^k)^T (Ax^k - b) \\ \geq \quad & (x^k - x)^T \nabla f(x) + \lambda^T (Ax - b) - (\lambda^k)^T (Ax^k - b). \end{aligned} \tag{1.10}$$

Since (x, λ) is a feasible solution of the dual problem and $x^k \in \mathcal{X}$, set $x' = x^k$ in (1.4), we obtain

$$(x^k - x)^T \nabla f(x) \geq (x^k - x)^T A^T \lambda = \lambda^T A(x^k - x).$$

Substituting it in the right hand side of (1.10), we obtain

$$\begin{aligned} & L(x^k, \lambda^k) - L(x, \lambda) \\ \geq \quad & \lambda^T A(x^k - x) + \lambda^T (Ax - b) - (\lambda^k)^T (Ax^k - b) \\ = \quad & (\lambda - \lambda^k)^T (Ax^k - b). \end{aligned}$$

The assertion of this lemma is proved. □

2 Dual ascent method

We assume that f is strictly convex and thus, for any given λ^k, the x-minimization problem (1.5a) has the unique solution x^k. Note that this assumption does not hold in many important applications, so dual ascent often cannot be used. As an example, if f is a nonzero affine function of any component of x, then the x-minimization (1.5a) fails, since

$L(x, \lambda)$ is unbounded below in x for most λ.

Dual Ascent Method(DAM)

Let (x^0, λ^0) be a pair of feasible solution of the dual problem (1.3).

For $k = 0, 1, \ldots$, do: Given dual feasible pair (x^k, λ^k), set

$$\lambda^{k+1} = \lambda^k - \beta_k(Ax^k - b), \tag{2.1a}$$

and let

$$x^{k+1} = \arg\min \{L(x, \lambda^{k+1}) \,\big|\, x \in \mathcal{X}\}. \tag{2.1b}$$

The parameter β_k is selected such that the dual feasible pairs (x^k, λ^k) and (x^{k+1}, λ^{k+1}) satisfy the condition

$$(\lambda^k - \lambda^{k+1})^T A(x^k - x^{k+1}) \leq \frac{\nu}{\beta_k}\|\lambda^k - \lambda^{k+1}\|^2, \ \nu \in (0, 1). \tag{2.1c}$$

Due to (1.8), by using Armijo-like line search to find a β_k to satisfy (2.1c), there is a $\beta_{\min} > 0$ such that

$$\inf_k\{\beta_k\} \geq \beta_{\min}.$$

Note that each pair (x^k, λ^k) generated by the dual ascent method is dual feasible.

Lemma 2.1 *Let* $\{(x^k, \lambda^k)\}$ *be the sequence generated by the dual ascent method* (2.1). *Then we have*

$$L(x^{k+1}, \lambda^{k+1}) - L(x^k, \lambda^k) \geq \frac{1-\nu}{\beta_k}\|\lambda^k - \lambda^{k+1}\|^2. \tag{2.2}$$

Proof. Since the sequence $\{(x^k, \lambda^k)\}$ is dual feasible, by setting $(x^k, \lambda^k) = (x^{k+1}, \lambda^{k+1})$ and $(x, \lambda) = (x^k, \lambda^k)$ in (1.9), we obtain

$$L(x^{k+1}, \lambda^{k+1}) - L(x^k, \lambda^k) \geq (\lambda^k - \lambda^{k+1})^T(Ax^{k+1} - b). \tag{2.3}$$

It follows from (2.1c) that

$$\begin{aligned}&(\lambda^k - \lambda^{k+1})^T(Ax^{k+1} - b)\\ &\quad \geq \quad (\lambda^k - \lambda^{k+1})^T(Ax^k - b) - \frac{\nu}{\beta_k}\|\lambda^k - \lambda^{k+1}\|^2.\end{aligned} \tag{2.4}$$

Note that from (2.1a) we have

$$(\lambda^k - \lambda^{k+1})^T(Ax^k - b) = \frac{1}{\beta_k}\|\lambda^k - \lambda^{k+1}\|^2. \tag{2.5}$$

The assertion of this lemma is proved. □

Lemma 2.2 *Let* $\{(x^k, \lambda^k)\}$ *be the sequence generated by the dual ascent method* (2.1). *Then for any* $\lambda^* \in \Lambda^*$, *we have*

$$\|\lambda^{k+1}-\lambda^*\|^2 \leq \|\lambda^k-\lambda^*\|^2+\|\lambda^k-\lambda^{k+1}\|^2-2\beta_k\big(L(x^*,\lambda^*)-L(x^k,\lambda^k)\big). \quad (2.6)$$

Proof. It follows from (2.1a) that

$$\begin{aligned}
&\|\lambda^{k+1} - \lambda^*\|^2 \\
&\quad = \|(\lambda^k - \lambda^*) - (\lambda^k - \lambda^{k+1})\|^2 \\
&\quad = \|\lambda^k - \lambda^*\|^2 + \|\lambda^k - \lambda^{k+1}\|^2 - 2(\lambda^k - \lambda^*)^T \beta_k (Ax^k - b). \qquad (2.7)
\end{aligned}$$

In addition, because (x^*, λ^*) is dual feasible, by setting $(x, \lambda) = (x^*, \lambda^*)$ in (1.9), we obtain

$$(\lambda^k - \lambda^*)^T (Ax^k - b) \geq L(x^*, \lambda^*) - L(x^k, \lambda^k). \qquad (2.8)$$

Substituting (2.8) in (2.7), the assertion follows directly. □

Lemma 2.3 *Let* $\{(x^k, \lambda^k)\}$ *be the sequence generated by the dual ascent method* (2.1).

Then for any $\lambda^* \in \Lambda^*$, we have

$$\begin{aligned} \|\lambda^{k+1} - \lambda^*\|^2 \quad \leq \quad & \|\lambda^k - \lambda^*\|^2 - (1-2\nu)\|\lambda^k - \lambda^{k+1}\|^2 \\ & -2\beta_k\big(L(x^*, \lambda^*) - L(x^{k+1}, \lambda^{k+1})\big). \end{aligned} \tag{2.9}$$

Thus the sequence $\{\lambda^k\}$ is Fejèr monotone with respect to Λ^* when $\nu \leq 1/2$.

Proof. First, it follows from (2.6) that

$$\begin{aligned} &\|\lambda^{k+1} - \lambda^*\|^2 \leq \|\lambda^k - \lambda^*\|^2 + \|\lambda^k - \lambda^{k+1}\|^2 \\ &-2\beta_k\big(L(x^*,\lambda^*) - L(x^{k+1},\lambda^{k+1})\big) + 2\beta_k\big(L(x^k,\lambda^k) - L(x^{k+1},\lambda^{k+1})\big). \end{aligned} \tag{2.10}$$

Using (2.2), we get

$$2\beta_k\big(L(x^k, \lambda^k) - L(x^{k+1}, \lambda^{k+1})\big) \leq -2(1-\nu)\|\lambda^k - \lambda^{k+1}\|^2.$$

Substituting it in (2.10), we get the assertion (2.9). The Lemma is proved. □

Theorem 2.1 *Let $\{(x^k, \lambda^k)\}$ be the sequence generated by the dual ascent method* (2.1). *If $\nu \leq 1/2$, the sequence $\{\lambda^k\}$ is Fejèr monotone with respect to Λ^*. Moreover,*

$$L(x^*, \lambda^*) - L(x^k, \lambda^k) \leq \frac{1}{2k\beta_{\min}}\|\lambda^0 - \lambda^*\|^2. \tag{2.11}$$

Proof. Because $L(x^*, \lambda^*) - L(x^{k+1}, \lambda^{k+1}) > 0$ and $\nu \leq 1/2$, it follows from (2.9) that

$$\|\lambda^{k+1} - \lambda^*\|^2 < \|\lambda^k - \lambda^*\|^2,$$

whenever $(x^k, \lambda^k) \notin \Omega^*$. Again, from (2.9) we obtain that

$$2\beta_{\min}\big(L(x^*, \lambda^*) - L(x^{l+1}, \lambda^{l+1})\big) \leq \|\lambda^l - \lambda^*\|^2 - \|\lambda^{l+1} - \lambda^*\|^2.$$

Summing the above inequality over $j = 0, \ldots, k-1$ and using the increasing property of $\{L(x^k, \lambda^k)\}$, the assertion (2.11) follows directly. □

3 Iterations complexity

Theorem 3.1 *Let $\{(x^k, \lambda^k)\}$ be the sequence generated by the dual ascent method* (2.1) *and $\beta_k \equiv \beta$. Then for any $k \geq 0$ and $(x^*, \lambda^*) \in \mathcal{X}^* \times \Lambda^*$, we have*

$$\begin{aligned} & 2k\beta\big(L(x^k, \lambda^k) - L(x^*, \lambda^*)\big) \\ \geq & \sum_{j=0}^{k-1}\Big(2j(1-\nu) + (1-2\nu)\Big)\|\lambda^j - \lambda^{j+1}\|^2 - \|\lambda^0 - \lambda^*\|^2. \end{aligned} \tag{3.1}$$

Proof. It follows from (2.9) that, for all $j \geq 0$, we have

$$\begin{aligned} & 2\beta(L(x^{j+1},\lambda^{j+1}) - L(x^*,\lambda^*)) \\ & \quad \geq \quad \|\lambda^{j+1} - \lambda^*\|^2 - \|\lambda^j - \lambda^*\|^2 + (1-2\nu)\|\lambda^j - \lambda^{j+1}\|^2. \end{aligned}$$

Using the fact $L(x^j,\lambda^j) - L(x^*,\lambda^*) < 0$, summing the above inequality over $j = 0,\ldots,k-1$, we obtain

$$\begin{aligned} & 2\beta\Big(\sum_{j=0}^{k-1} L(x^{j+1},\lambda^{j+1}) - kL(x^*,\lambda^*)\Big) \\ & \quad \geq \quad \|\lambda^k - \lambda^*\|^2 - \|\lambda^0 - \lambda^*\|^2 + (1-2\nu)\sum_{j=0}^{k-1}\|\lambda^j - \lambda^{j+1}\|^2. \end{aligned} \tag{3.2}$$

By using Lemma 2.1 for $k = j$, we get

$$\beta(L(x^{j+1},\lambda^{j+1}) - L(x^j,\lambda^j)) \geq (1-\nu)\|\lambda^j - \lambda^{j+1}\|^2.$$

Multiplying the last inequality by $2j$ and summing over $j = 0, \ldots, k-1$, it follows that

$$\begin{aligned} & 2\beta \sum_{j=0}^{k-1} \Big((j+1) L(x^{j+1}, \lambda^{j+1}) - L(x^{j+1}, \lambda^{j+1}) - j L(x^j, \lambda^j) \Big) \\ & \geq \quad \sum_{j=0}^{k-1} 2j(1-\nu) \|\lambda^j - \lambda^{j+1}\|^2, \end{aligned}$$

which simplifies to

$$2\beta \Big(k L(x^k, \lambda^k) - \sum_{j=0}^{k-1} L(x^{j+1}, \lambda^{j+1}) \Big) \geq \sum_{j=0}^{k-1} 2j(1-\nu) \|\lambda^j - \lambda^{j+1}\|^2. \quad (3.3)$$

Adding (3.2) and (3.3), we get

$$\begin{aligned} & 2k\beta \big(L(x^k, \lambda^k) - L(x^*, \lambda^*) \big) \\ & \geq \quad \sum_{j=0}^{k-1} \Big(2j(1-\nu) + (1-2\nu) \Big) \|\lambda^j - \lambda^{j+1}\|^2 - \|\lambda^0 - \lambda^*\|^2. \end{aligned}$$

The proof is complete. $\square$

♣ In fact, for any $\nu \in (0,1)$, the iteration-complexity of the single projected gradient method is $O(1/k)$. For example, if $\nu = 0.9$, then

$$(1-2\nu) + 2j(1-\nu) \geq 0, \ \forall\, j \geq 4.$$

In this case, it follows from (3.1) that

$$\begin{aligned} & 2k\beta\big(L(x^*,\lambda^*) - L(x^k,\lambda^k)\big) \\ & \quad \leq \quad \|\lambda^0 - \lambda^*\|^2 + \textstyle\sum_{l=0}^{3}\Big((2\nu-1) + 2l(\nu-1)\Big)\|\lambda^j - \lambda^{j+1}\|^2. \end{aligned} \tag{3.4}$$

Since $\nu \leq 0.9$, we have

$$L(x^*,\lambda^*) - L(x^k,\lambda^k) \leq \frac{1}{2k\beta}\Big(\|\lambda^0 - \lambda^*\|^2 + \tfrac{4}{5}\textstyle\sum_{l=0}^{3}\|\lambda^j - \lambda^{j+1}\|^2\Big). \tag{3.5}$$

Practical Condition In practical computation, instead of the condition (2.1c), we use

$$\|\beta_k A(x^k - x^{k+1})\| \leq \nu\|\lambda^k - \lambda^{k+1}\|, \quad \nu \in (0,1) \tag{3.6}$$

as the acceptance condition. It is clear that above condition is stronger than the condition (2.1c). We use the following self-adaptive dual ascent method:

Self-adaptive dual ascent method:

Step 0. Set $\beta_0 = 1, \nu \in (0,1), \lambda^0 \in \Re^m, x^0 = \arg\min\{L(x,\lambda^0)|x \in \mathcal{X}\}$.

For $k = 0, 1, \ldots$, if the stopping criterium is not satisfied, do:

Step 1. $\tilde{\lambda}^k = \lambda^k - \beta_k(Ax^k - b), \quad \tilde{x}^k = \arg\min\{L(x,\tilde{\lambda}^k) \mid x \in \mathcal{X}\},$

$r_k := \|\beta_k A(x^k - \tilde{x}^k)\|/\|\lambda^k - \tilde{\lambda}^k\|,$

while $r_k > \nu$

$\beta_k := \frac{2}{3}\beta_k * \min\{1, \frac{1}{r_k}\},$

$\tilde{\lambda}^k = \lambda^k - \beta_k(Ax^k - b), \quad \tilde{x}^k = \arg\min\{L(x,\tilde{\lambda}^k) \mid x \in \mathcal{X}\},$

$r_k := \|\beta_k A(x^k - \tilde{x}^k)\|/\|\lambda^k - \tilde{\lambda}^k\|,$

end(while)

$\lambda^{k+1} = \tilde{\lambda}^k,$

If $r_k \le \mu$ then $\beta_k := \beta_k * 1.5,$ **end(if)**

Step 2. $\beta_{k+1} = \beta_k$ and $k = k + 1,$ go to Step 1.

采用上述程序但略去 **If** $r_k \le \mu$ then $\beta_k := \beta_k * 1.5$ **end(if)** 的做法, 将大大增加迭代步数, 有时甚至导致计算失败.

4 Applications of the self-adaptive dual-ascent method

在统计学中, 一个对角元均为 1 的对称半正定矩阵称为相关性矩阵 (Correlation Matrix). 对给定的对称矩阵 C, 求 F-模下与 C 距离最近的相关性矩阵, 其数学表达式是

$$\min\{\frac{1}{2}\|X-C\|_F^2 \mid \mathrm{diag}(X)=e,\ X\in S_+^n\}, \tag{4.1}$$

其中 e 表示每个分量都为 1 的 n-维向量, S_+^n 表示 $n\times n$ 正半定锥的集合. 问题 (4.1) 是形如 (1.1) 的等式约束凸优化问题, 其中 $\|A^TA\|=1$.

我们用 $z\in\Re^n$ 作为等式约束 $\mathrm{diag}(X)=e$ 的 Lagrange 乘子.

用第四讲的 PPA 算法求解问题 (4.1), 具体做法可见第四讲的 § 4.1.

对给定的 z^k, 产生 $\tilde{X}^k$ 的方法是:

$$\min\{\frac{1}{2}\|X-C\|_F^2-(z^k)^T(\mathrm{diag}(X)-e)|X\in S_+^n\}. \tag{4.2}$$

子问题 (4.2) 求解的具体做法: 化为等价问题

$$\min\{\frac{1}{2}\|X-(C+\mathrm{diag}(z^k))\|_F^2 | X\in S_+^n\}.$$

因此我们只要考虑如何求解

$$\tilde{X}^k = \mathrm{Argmin}\{\frac{1}{2}\|X-A\|_F^2 \mid X\in S_+^n\}. \tag{4.3}$$

问题 (4.3) 的解法在第四讲里已经作了介绍.

采用 **Dual Ascent Method**, 每步迭代中最大的花费是要对给定的 λ^k, 生成一个 Dual feasible pair (x^k, λ^k), 其中

$$x^k = \mathrm{Argmin}\{L(x,\lambda^k)|x\in\mathcal{X}\}.$$

这个子问题的形式就是 (4.2), 我们只是将它的解及成 X^k.

我们对不同的方法进行对比计算. 在 Matlab 程序中, 对称矩阵特征值分解, 都使用 mexeig 子程序. 试验结果表明, 对这一类问题, **Dual Ascent Method** 比第四讲的 **Customized PPA** 还要快一倍左右.

Code 5.A. Matlab code for Creating the test examples

```
%%%%    DEMO        Comparison the two different methods %%%%%%%%%%%%%%
%%%  min { (1/2)|X-C|^2 | X is positive semi-definite, X_{jj}=1 }  %%%
clear; close all; clc;                                              %(1)
n = 500;            tol=1e-6;                                       %(2)
%% Generating the given matrix C                                    %(3)
rand('state',0);          randn('state',0);                         %(4)
C=rand(n,n);               C=(C'+C) - ones(n,n) + eye(n);           %(5)
%% C is symmetric, C_{ij} in (-1,1) for i\ne j,  C_{jj} in (0,2)   %(6)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%     %(7)
%% Run  Extende PPA with mexeig %%                                  %(8)
   r = 2.0;   s = 1.01/r;    gamma = 1.5;     %% Given Parameter    %(9)
   PPA_G(n,C,r,s,tol,gamma)                                        %(10)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%    %(11)
 %% Run Dual-Ascent Method                                         %(12)
   beta=1.0;                                 %% Given Parameter  %(13)
   Dual_A(n,C,beta,tol)                                            %(14)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

生成的对称矩阵 C, 对角元在 $(0, 2)$ 之间, 非对角元在 $(-1, 1)$ 之间.

Code 5.1 Matlab Code of the Extended PPA

```
%%%     Extended PPA for calibrating correlation matrix                %(1)
function PPA_G(n,C,r,s,tol,gamma)                                      %(2)
X = eye(n);       y = zeros(n,1);    tic;  %% The initial iterate     %(3)
stopc=1;        k=0;                                                   %(4)
while (stopc>tol && k<=100)           %% Beginning of an Iteration    %(5)
    if mod(k,1)==0; fprintf('k=%3d  epsm=%9.3e\n',k,stopc);  end;      %(6)
    X0 = X;      y0 = y;             k=k+1;                            %(7)
    yt = y0 - (diag(X0)-ones(n,1))/s;                                  %(8)
    A =(X0*r + C + diag(yt*2-y0))/(1+r);                               %(9)
     [V,D] = mexeig(A);   D = max(0,D);  XT = (V*D)*V'; %% mexeig     %(10)
     EX = X0-XT;          EY = y0-yt;                                  %(11)
     ex = max(max(abs(EX)));  ey = max(abs(EY));                       %(12)
     stopc= max(ex,ey);                                                %(13)
    X = X0-EX*gamma;          y = y0-EY*gamma;                         %(14)
end                                        %% End of an Iteration     %(15)
toc                                                                    %(16)
TB = max(abs(diag(X-eye(n))));                                         %(17)
 fprintf('k=%3d epsm=%9.3e  TB=%8.5f\n\n',k,stopc,TB);                 %(18)
```

Code 5.2 Matlab Code of Dual Ascent Method

```
%%%   Dual-Ascent-Method, Dual variable z                              %(1)
function Dual_A(n,C,beta,tol)                                          %(2)
 z=zeros(n,1);      tic;            %%  The initial iterate           %(3)
 A= C + diag(z);  [V,D]=mexeig(A);  D=max(0,D);    X=(V*D)*V';         %(4)
  r=1;   k=0;     l=0;     stopc=1;                                     %(5)
while (stopc>tol && k<=60)        %% Beginning of an Iteration         %(6)
     gz= diag(X)-ones(n,1);       stopc=max(abs(gz));                  %(7)
        k= k+1;        l=l+1;                                          %(8)
        dz=gz*beta;       zt=z-dz;        A= C + diag(zt);           %(9)
        [V,D]=mexeig(A);  D=max(0,D);     XT=(V*D)*V';              %(10)
        df=(diag(X)-diag(XT))*beta;       r=norm(df)/norm(dz);      %(11)
        while r>0.9                                                  %(12)
          beta=0.8*beta;   l=l+1;                                    %(13)
          dz=gz*beta;     zt=z-dz;       A= C + diag(zt);           %(14)
          [V,D]=mexeig(A);  D=max(0,D);   XT=(V*D)*V';              %(15)
          df=(diag(X)-diag(XT))*beta;     r=norm(df)/norm(dz);      %(16)
        end;                                                         %(17)
        z = zt;   X=XT;     if r <0.6    beta=beta*1.5;  end;        %(18)
 end;                                  %% End of an Iteration        %(19)
toc;     fprintf('  k=%4d   epsm=%9.3e   l=%4d \n',k,stopc,l);      %(20)
```

矩阵校正问题 (4.1)　　精度要求 $\varepsilon = 10^{-4}$

$n \times n$ Matrix	Extended PPA		Dual-Ascent Method		
$n =$	No. It	CPU Sec.	No. It	No. of Solving Sub-Prob (2.1b)	CPU Sec.
100	18	0.11	11	11	0.07
200	21	0.36	12	12	0.21
500	22	3.41	12	12	1.80
800	24	12.65	13	13	7.05
1000	25	24.75	13	13	12.51
1500	30	93.74	13	13	42.42
2000	34	241.25	14	14	103.85

The dual ascent method converges much faster than the extended PPA.

$$\frac{\text{CPU. Time of the dual ascent method}}{\text{CPU. Time of the extended PPA}} \leq \begin{cases} 55\% & n < 1000 \\ 45\% & n \geq 1000 \end{cases}$$

矩阵校正问题 (4.1)　　精度要求 $\varepsilon = 10^{-6}$

$n \times n$ Matrix	Extended PPA		Dual-Ascent Method		
$n =$	No. It	CPU Sec.	No. It	No. of Solving Sub-Prob (2.1b)	CPU Sec.
100	26	0.16	14	14	0.09
200	29	0.50	17	17	0.29
500	32	4.96	17	17	2.54
800	35	18.45	19	19	10.29
1000	36	35.64	17	17	18.75
1500	44	137.50	18	18	58.74
2000	50	354.78	20	20	148.36

The dual ascent method converges much faster than the extended PPA.

$$\frac{\text{CPU. Time of the dual ascent method}}{\text{CPU. Time of the extended PPA}} \leq \begin{cases} 55\% & n < 1000 \\ 45\% & n \geq 1000 \end{cases}$$

5 An accelerated two-steps dual ascent method

According to the basic idea of Nesterov [6], we can construct the accelerated two-steps dual ascent method. Besides $\{\lambda^k\}$, it generates an auxiliary sequence $\{\eta^k\}$.

A two-steps dual ascent method

Step 0. Take $\beta > 0$, $\lambda^1 \in R^n$. Set $\eta^1 = \lambda^1$, $t_1 = 1$.

Step k. ($k \geq 1$) With given (λ^k, η^k), produce the dual feasible pair (x_η^k, η^k) and let

$$\lambda^{k+1} = \eta^k - \beta_k(Ax_\eta^k - b), \tag{5.1a}$$

then generate the new dual feasible pair (x^{k+1}, λ^{k+1}). The step size β_k should ensure the two dual feasible pairs, (x_η^k, η^k) and (x^{k+1}, λ^{k+1}), to satisfy

$$(\eta^k - \lambda^{k+1})^T A(x_\eta^k - x^{k+1}) \leq \frac{1}{2\beta_k}\|\eta^k - \lambda^{k+1}\|^2. \tag{5.1b}$$

Set

$$\eta^{k+1} = \lambda^{k+1} + \Big(\frac{t_k - 1}{t_{k+1}}\Big)(\lambda^{k+1} - \lambda^k), \tag{5.1c}$$

where

$$t_{k+1} = \frac{1 + \sqrt{1 + 4t_k^2}}{2}. \tag{5.1d}$$

The method is called two-steps dual ascent method because each iteration consists of two steps. The k-th iteration begins with (λ^k, η^k), the first step (5.1a) produces λ^{k+1} and the second one (5.1c) updates η^{k+1}. In each iteration, it needs at least to produce two dual feasible pairs, namely

$$(x_\eta^k, \eta^k) \qquad \text{and} \qquad (x^{k+1}, \lambda^{k+1}).$$

It is assumed that the positive sequence $\{\beta_k\}$ is non- increasing. We show that the proposed two-steps dual ascent method is convergent with the iteration-complexity $O(1/k^2)$. The proof is similar as those in [1].

Lemma 5.1 *Let* λ^{k+1} *be given by* (5.1a) *and the step size condition* (5.1b) *be satisfied. Then we have*

$$\begin{aligned} & 2\beta_k(L(x^{k+1}, \lambda^{k+1}) - L(x, \lambda)) \\ & \quad \geq \quad \|\eta^k - \lambda^{k+1}\|^2 + 2(\lambda^{k+1} - \eta^k)^T(\eta^k - \lambda), \ \forall\, \lambda \in \Re^m. \end{aligned} \tag{5.2}$$

Proof. By using (1.9), for any feasible solution (x, λ) of the dual problem (1.3), we get

$$L(x_\eta^k, \eta^k) - L(x, \lambda) \geq (\lambda - \eta^k)^T(Ax_\eta^k - b).$$

Due to (5.1a), we have

$$\begin{aligned}(\lambda-\eta^k)^T(Ax_\eta^k-b) &= \{(\lambda-\lambda^{k+1})+(\lambda^{k+1}-\eta^k)\}^T(Ax_\eta^k-b)\\ &= \frac{1}{\beta_k}(\lambda-\lambda^{k+1})(\eta^k-\lambda^{k+1})-\frac{1}{\beta_k}\|\eta^k-\lambda^{k+1}\|^2,\end{aligned}$$

and consequently

$$L(x_\eta^k,\eta^k)-L(x,\lambda)\geq\frac{1}{\beta_k}(\lambda-\lambda^{k+1})(\eta^k-\lambda^{k+1})-\frac{1}{\beta_k}\|\eta^k-\lambda^{k+1}\|^2. \quad (5.3)$$

Again, by setting $k:=k+1$ and $(x,\lambda)=(x_\eta^k,\eta^k)$ in (1.9), we obtain

$$\begin{aligned}&L(x^{k+1},\lambda^{k+1})-L(x_\eta^k,\eta^k)\\ &\quad\geq (\eta^k-\lambda^{k+1})^T(Ax^{k+1}-b)\\ &\quad= (\eta^k-\lambda^{k+1})^T\{(Ax_\eta^k-b)-A(x_\eta^k-x^{k+1})\}\\ &\quad= \frac{1}{\beta_k}\|\eta^k-\lambda^{k+1}\|^2-(\eta^k-\lambda^{k+1})^TA(x_\eta^k-x^{k+1})\\ &\quad\geq \frac{1}{2\beta_k}\|\eta^k-\lambda^{k+1}\|^2. \qquad (5.4)\end{aligned}$$

Adding (5.3) and (5.4),

$$\begin{aligned} & L(x^{k+1},\lambda^{k+1}) - L(x,\lambda) \\ \geq\ & \frac{1}{\beta_k}(\lambda-\lambda^{k+1})(\eta^k-\lambda^{k+1}) - \frac{1}{2\beta_k}\|\eta^k-\lambda^{k+1}\|^2 \\ =\ & \frac{1}{\beta_k}(\lambda-\eta^k)^T(\eta^k-\lambda^{k+1}) + \frac{1}{2\beta_k}\|\eta^k-\lambda^{k+1}\|^2. \end{aligned} \tag{5.5}$$

The above inequality can be rewritten as (5.2) and the lemma is proved. □

To derive the iteration-complexity of the two-steps projected gradient method, we need to prove some properties of the corresponding sequence.

Lemma 5.2 *The sequences* $\{\lambda^k\}$ *and* $\{\eta^k\}$ *generated by the proposed two-steps dual ascent method satisfy*

$$2\beta_k t_k^2 v_k - 2\beta_{k+1} t_{k+1}^2 v_{k+1} \geq \|u^{k+1}\|^2 - \|u^k\|^2,\ \forall k \geq 1, \tag{5.6}$$

where $v_k := L(x^*,\lambda^*) - L(x^{k+1},\lambda^{k+1})$ *and* $u^k := t_k\lambda^{k+1} - (t_k-1)\lambda^k - \lambda^*$.

Proof. By using Lemma 5.1 for $k+1$, $x = \lambda^{k+1}$ and $x = \lambda^*$ we get

$$\begin{aligned} & 2\beta_{k+1}\big(L(x^{k+2}, \lambda^{k+2}) - L(x^{k+1}, \lambda^{k+1})\big) \\ & \quad \geq \quad \|\eta^{k+1} - \lambda^{k+2}\|^2 + 2(\lambda^{k+2} - \eta^{k+1})^T(\eta^{k+1} - \lambda^{k+1}), \end{aligned}$$

and

$$\begin{aligned} & 2\beta_{k+1}\big(L(x^{k+2}, \lambda^{k+2}) - L(x^*, \lambda^*)\big) \\ & \quad \geq \quad \|\eta^{k+1} - \lambda^{k+2}\|^2 + 2(\lambda^{k+2} - \eta^{k+1})^T(\eta^{k+1} - \lambda^*). \end{aligned}$$

Using the definition of v_k, we get

$$2\beta_{k+1}(v_k - v_{k+1}) \geq \|\eta^{k+1} - \lambda^{k+2}\|^2 + 2(\lambda^{k+1} - \eta^{k+1})^T(\eta^{k+1} - \lambda^{k+2}), \quad (5.7)$$

and

$$-2\beta_{k+1}v_{k+1} \geq \|\eta^{k+1} - \lambda^{k+2}\|^2 + 2(\lambda^* - \eta^{k+1})^T(\eta^{k+1} - \lambda^{k+2}). \quad (5.8)$$

To get a relation between v_k and v_{k+1}, we multiply (5.7) by $(t_{k+1}-1)$ and add it to (5.8):

$$\begin{aligned} &2\beta_{k+1}\big((t_{k+1}-1)v_k - t_{k+1}v_{k+1}\big) \\ &\quad\geq\quad t_{k+1}\|\lambda^{k+2}-\eta^{k+1}\|^2 \\ &\qquad\quad +2(\lambda^{k+2}-\eta^{k+1})^T\big(t_{k+1}\eta^{k+1}-(t_{k+1}-1)\lambda^{k+1}-\lambda^*\big). \end{aligned}$$

Multiplying the last inequality by t_{k+1} and using

$$t_k^2 = t_{k+1}^2 - t_{k+1} \qquad \big(\text{and thus} \quad t_{k+1} = (1+\sqrt{1+4t_k^2})/2 \quad \text{as in} \quad (5.1\text{d})\big),$$

which yields

$$\begin{aligned} &2\beta_{k+1}\big(t_k^2 v_k - t_{k+1}^2 v_{k+1}\big) \\ &\quad\geq\quad \|t_{k+1}(\lambda^{k+2}-\eta^{k+1})\|^2 \\ &\qquad\quad +2t_{k+1}(\lambda^{k+2}-\eta^{k+1})^T\big(t_{k+1}\eta^{k+1}-(t_{k+1}-1)\lambda^{k+1}-\lambda^*\big). \end{aligned}$$

Applying the relation

$$\|a-b\|^2 + 2(a-b)^T(b-c) = \|a-c\|^2 - \|b-c\|^2$$

to the right-hand side of the last inequality with

$$a := t_{k+1}\lambda^{k+2}, \qquad b := t_{k+1}\eta^{k+1}, \qquad c := (t_{k+1}-1)\lambda^{k+1} + \lambda^*,$$

and using the fact $2\beta_k t_k^2 v_k \geq 2\beta_{k+1} t_k^2 v_k$ (since $\{\beta_k\}$ is non-increasing), we get

$$\begin{aligned} & 2\beta_k t_k^2 v_k - 2\beta_{k+1} t_{k+1}^2 v_{k+1} \\ \geq \quad & \|t_{k+1}\lambda^{k+2} - (t_{k+1}-1)\lambda^{k+1} - \lambda^*\|^2 \\ & -\|t_{k+1}\eta^{k+1} - (t_{k+1}-1)\lambda^{k+1} - \lambda^*\|^2. \end{aligned}$$

In order to write the above inequality in the form (5.6) with

$$u^k = t_k\lambda^{k+1} - (t_k - 1)\lambda^k - \lambda^*,$$

we need only to set

$$t_{k+1}\eta^{k+1} - (t_{k+1}-1)\lambda^{k+1} - \lambda^* = t_k\lambda^{k+1} - (t_k-1)\lambda^k - \lambda^*.$$

From the last equality we obtain

$$\eta^{k+1} = \lambda^{k+1} + \Big(\frac{t_k - 1}{t_{k+1}}\Big)(\lambda^{k+1} - \lambda^k).$$

This is just the form (5.1c) in the accelerated two-steps version of the dual ascent method

and the lemma is proved. $\square$

To proceed the proof of the main theorem, we need the following Lemma 5.3 and Lemma 5.4, which have also been considered in [1]. We omit their proofs as they are trivial.

Lemma 5.3 *Let* $\{a_k\}$ *and* $\{b_k\}$ *be positive sequences of reals satisfying*

$$a_k - a_{k+1} \geq b_{k+1} - b_k \quad \forall\, k \geq 1.$$

Then, $a_k \leq a_1 + b_1$ *for every* $k \geq 1$.

Lemma 5.4 *The positive sequence* $\{t_k\}$ *generated by*

$$t_{k+1} = \frac{1 + \sqrt{1 + 4t_k^2}}{2}, \qquad \textit{with} \qquad t_1 = 1$$

satisfies

$$t_k \geq \frac{k+1}{2}, \quad \forall\, k \geq 1.$$

Now, we are ready to show that the proposed two-steps projected gradient method is convergent with the rate $O(1/k^2)$.

Theorem 5.1 *Let* $\{\lambda^k\}$ *and* $\{\eta^k\}$ *be generated by the proposed two-steps dual ascent*

method . Then, for any $k \geq 1$, we have

$$L(x^*, \lambda^*) - L(x^k, \lambda^k) \leq \frac{2\|\lambda^1 - \lambda^*\|^2}{\beta_k k^2}, \ \ \forall\, \lambda^* \in \Omega^*. \tag{5.9}$$

Proof. Let us define the quantities

$$a_k := 2\beta_k t_k^2 v_k, \qquad b_k := \|u^k\|^2.$$

By using Lemma 5.2 and Lemma 5.3, we obtain

$$2\beta_k t_k^2 v_k \leq a_1 + b_1,$$

which combined with the definition v_k and $t_k \geq (k+1)/2$ (by Lemma 5.4) yields

$$L(x^*, \lambda^*) - L(x^{k+1}, \lambda^{k+1}) = v_k \leq \frac{2(a_1 + b_1)}{\beta_k (k+1)^2} \leq \frac{2(a_1 + b_1)}{\beta_{k+1}(k+1)^2}. \tag{5.10}$$

Since $t_1 = 1$, and using the definition of u_k given in Lemma 5.2, we have

$$a_1 = 2\beta_1 t_1^2 v_1 = 2\beta_1 v_1 = 2\beta_1 \big(L(x^*, \lambda^*) - L(x^2, \lambda^2)\big),$$

and

$$b_1 = \|u^1\|^2 = \|\lambda^2 - \lambda^*\|^2.$$

Setting $\lambda = \lambda^*$ and $k = 1$ in (5.2), we have

$$\begin{aligned} 2\beta_1(L(x^*, \lambda^*) - L(x^2, \lambda^2)) &\leq 2(\eta^1 - \lambda^*)^T(\eta^1 - \lambda^2) - \|\eta^1 - \lambda^2\|^2 \\ &= \|\eta^1 - \lambda^*\|^2 - \|\lambda^2 - \lambda^*\|^2. \end{aligned}$$

Therefore, we have

$$\begin{aligned} a_1 + b_1 &= 2\beta_1(L(x^*, \lambda^*) - L(x^2, \lambda^2)) + \|\lambda^2 - \lambda^*\|^2 \\ &\leq \|\eta^1 - \lambda^*\|^2 - \|\lambda^2 - \lambda^*\|^2 + \|\lambda^2 - \lambda^*\|^2 \\ &= \|\lambda^1 - \lambda^*\|^2. \end{aligned}$$

Substituting it in (5.10), the assertion is proved. □

Based on Theorem 5.1, for obtaining an ε-optimal dual solution (denoted by λ) in the sense that $L(x^*, \lambda^*) - L(x, \lambda) \leq \varepsilon$, the number of iterations required by the proposed two- steps dual ascent method is at most $\lceil C/\sqrt{\varepsilon} - 1 \rceil$ where $C = 2\|\lambda^1 - \lambda^*\|^2/\beta$.

6 Conclusion remarks

According to my limited numerical experiences, it is very important to adjust the parameter β in the self-adaptive dual ascent method in Section 3. A suitable small β in (2.1a) will ensure the condition (2.1c) and the convergence. However, if

$$r_k = \frac{\|\beta_k A(x^k - \tilde{x}^k)\|}{\|\lambda^k - \tilde{\lambda}^k\|} \leq \mu, \quad (\text{say } \mu = 0.4)$$

the parameter β should be to enlarge for the trial in the next iteration.

Notice that, in the convergence rate proof of the accelerated two-steps dual ascent method, it is assumed that the nonnegative positive sequence $\{\beta_k\}$ is non-increasing. This “non-increasing” assumption will destroy the convergence behaviours in the practical computation.

References

[1] A. Beck and M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems, SIAM J. Imaging Science, 2 (2009), pp. 183-202.

[2] S. Boyd, N. Parikh, E. Chu, B. Peleato and J. Eckstein, Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers, Foundations and Trends in Machine Learning, 3 (2010), pp. 1-122.

[3] B. S. He, A class of projection and contraction methods for monotone variational inequalities, *Applied Mathematics and optimization*, 35: pp. 69–76, 1997.

[4] B. S. He and L. Z. Liao, Improvements of some projection methods for monotone nonlinear variational inequalities. *Journal of Optimization Theory and Applications* **112** (2002) 111-128.

[5] A. S. Nemirovsky and D. B. Yudin, *Problem Complexity and Method Efficiency in Optimization*, Wiley-Interscience Series in Discrete Mathematics, John Wiley $\&$ Sons, New York, 1983.

[6] Y. E. Nesterov, A method for solving the convex programming problem with convergence rate $O(1/k^2)$, Dokl. Akad. Nauk SSSR, 269 (1983), pp. 543-547.

[7] Y. E. Nesterov, Gradient methods for minimizing composite objective function, CORE report 2007; available at http://www.ecore.be/DPs/dp-1191313936.pdf.

[8] M.J.D. Powell, The Lagrange method and SAO with bounds on the dual variables, Dec. 2011, http://www.optimization-online.org/DB_FILE/2011/12/3295.pdf

[9] R.T. Rockafellar, Monotone operators and the proximal point algorithm, *SIAM J. Control Optimization*, 14: pp. 877-898, 1976.